Lab Assignment #5

INTRODUCTION TO DE2 SYSTEM INTERFACES

1. OBJECTIVES

- Becoming familiar with the system interface to seven segment LED displays on the DE2 protoboard,
- learning the process of encoding the 128 character set for displaying on the LCD panel on the DE2 protoboard,
- becoming familiar with the system interface to the LCD panel display on the DE2 protoboard,
- exercising the implementation process on the "terasic" protoboard DE2.

2. NE-1026 PRELAB ASSIGNMENT

Use the NE-1026 Computer Lab to log on to your Engineering College Computing (**ECC**) account, and complete the Prelab Assignment. For the later work on Lab Assignment #4 use the NE-2036 FPGA Lab and log on to your FPGA Lab accounts. The work on this prelab assignment will require access to the Altera Quartus-II Interactive Tutorials, and the "terasic" DE2 FPGA board tutorials, which are available on the Engineering College Computing (**ECC**) file server. All results of the Prelab Assignment work must be accounted for in the text of the Prelab Assignment Report, which is a prerequisite for, and due at the beginning time of the work in the FPGA Lab session.

2.1 CREATING THE DIRECTORY FOR LAB ASSIGNMENT #4 IN THE ECC ACCOUNT

After logging in to an ECC general account, open a new Terminal window by executing:

RMB → Background → Tools → Terminal

In the opened terminal window execute:

```
tcsh
source /eng/applications/altera/altera.csh
cd ~/fpga/lab
mkdir lab5
cd lab5
```

to create a directory named ~/fpga/lab/lab5, and make it the working directory in which the rest of this prelab assignment will be completed.

2.2 ACCESSING QUARTUS-II ENVIRONMENT

FPGA vendor Altera provides an application named Quartus II, which is its version of a "FPGA design and implementation environment" equivalent to Xilinx's ISE. For its Quartus II application Altera provides interactive and static tutorials. Quartus-II interactive tutorials are accessible on the ECC server at:

/eng/applications/altera/quartus

To start the interactive tutorials execute,

quartus

after which nothing happens for about two minutes. At the end of those two minutes the Quartus window pops up, and on top of it a small dialog box with the option to create a new project now, for which select

No

2.3 STUDYING THE LCD PANEL DATA SHEET

The LCD panel data sheets are accessible on the ECC server in the file:

/eng/applications/altera/DE2_System_v1.6/Datasheets/LCD/CFAH1602BTMCJP.pdf

Review carefully the sections which will be needed as a reference when preparing the VHDL code that describes the module BCD2LCDP shown in Figure 2.1:

- section 11 which describes the addressing of the character encoding ROM patterns,
- section 12 which describes the encoding of the eleven instructions which manipulate the behavior of the LCD panel display.

Studying the section 10 is recommended but not required.

2.4 DECODING THE INITIALIZATION INSTRUCTIONS FROM THE LCD_TEST.V FILE

The Verilog file **LCD_TEST.V** is accessible on the ECC server at:

/eng/applications/altera/DE2_System_v1.6/DE2_Demonstrations/DE2_Default

Study the five initialization instructions of the CASE structure which implements the LUT_INDEX table Use the Data Sheet studied in section 2.4 to discover what each of these instruction does because four of them will be implemented unchanged in the VHDL code that decibels the module BCD2LCDP.

2.5 STUDYING THE LCD_CONTROLLER.VHD FILE

The VHDL file **LCD_CONTROLLER.VHD** is accessible on the course webpage. Download the file to the Lab5 directory and study the VHDL code to understand what the module is doing because the module BCD2LCDP will need to use its instantiation. In short, **LCD_CONTROLLER** is providing some basic signaling/stimulus which is need for the functioning of the LCD display on the DE2 protoboard.

2.6 VHDL CODE FOR THE EXPERIMENT

Prepare the file named **main_lab5.vhd** containing your own VHDL code which describes the logic circuit architecture shown in Figure 2.1.

Components described by behavioral source code should provide for the following functions:

1. implementing UP/DOWN, two digit BCD counter with asynchronous RESET input shown in Figure 2.1;
2. implementing the BCD to LCD Panel decoder/driver module BCD2LCDP which should provide the following functions:
   - provide commands for initializing the LCD Panel,

- convert the BCD encoded digits to LCDP encoded characters.

Hint#1: The file **LCD_TEST.v,** which was studied in section 2.5, can provide an example for building the code for BCD2LCDP module. Still be aware of some differences:

- **LCD_TEST.v** code does not refresh the display, and that refreshing will be needed in BCD2LCDP after each increment of the counter,
- **LCD_TEST.v** code implements a two-line display, while this Lab assignment requires only one line display.

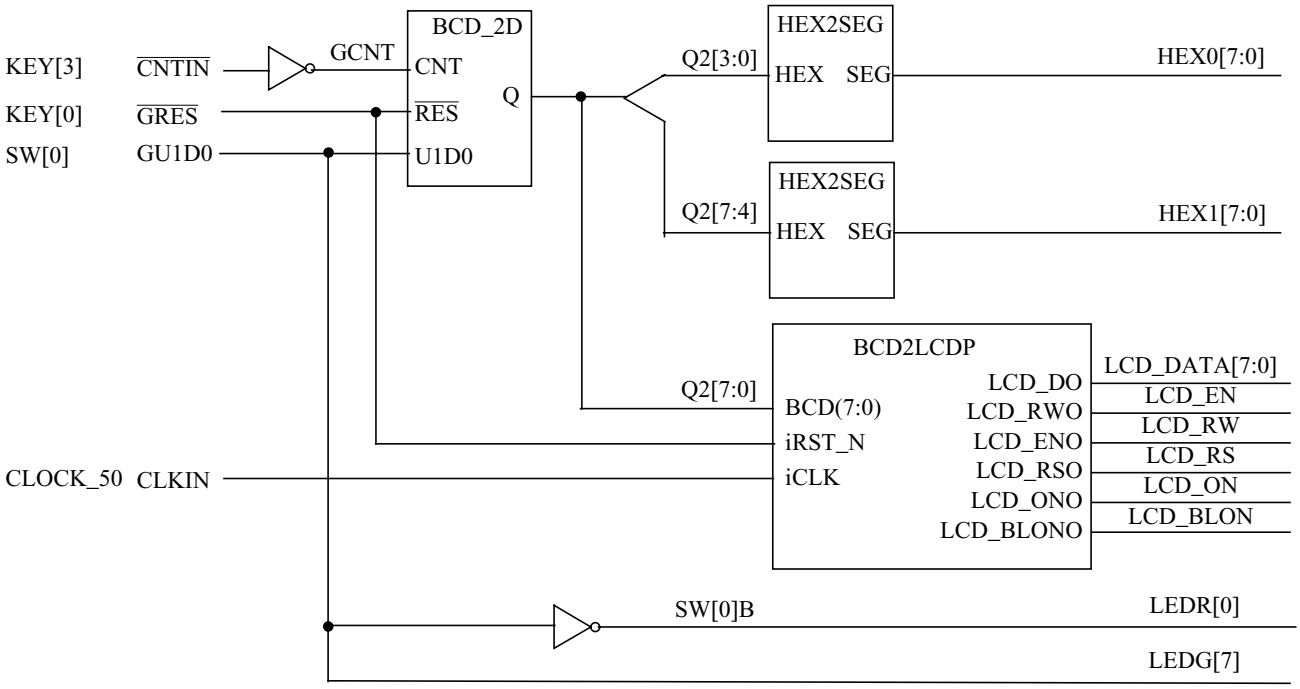

Figure 2.1 Architecture of the logic circuit for the experimentation with displays on DE2 FPGA protoboard.

Hint#2: When making pin assignments, refer to DE2 User Manual which is available in the file:

/eng/applications/altera/DE2_System_v1.6/DE2_user_manual/DE2_user_manual.pdf

The structural part of the code contained in the file named main_lab5.vhd should provide for the following functions:

1. connect the output signals of BCD counter to the inputs of two hexadecimal to seven segment code converters (HEX2SEG) by slicing the output signal into two slices of four least and four most significant bits,
2. designate the output signals of the two HEX2SEG converters exactly as shown in Figure 2.1, so that they carry the indices which correspond to the signals on the four digit seven segment module.
3. connect the reset signal $\overline{\text{GRES}}$ to the key (push button) KEY[0] signal;
4. connect the input CNTIN of the BCD counter to the key (push button) KEY[3] signal;
5. connect the input U1D0 of the BCD counter to the switch SW[0] signal;
6. connect the CLKIN signal to the 50 MHz input signal CLOCK_50;
7. connect the BCD2LCDP module to the output signals exactly as shown in Figure 2.1;

8. display zero on HEX2 and HEX3 seven segment display unitsl.
9. To simplify the simulation process, create a Test Bench module by,
   - copy the test bench template used in Lab#3 assignment to create the top module testb_lab5.vhd,
   - write a behavioral VHDL source code which will generate the sequences of all input signals which will show that the architecture in Figure 2.1:
     - can display the counter's state in parallel on both, the seven segment display and the LCD panel,
     - the BCD conuter can count up from 0 to 99,
     - the BCD conuter can count down from 99 to 0;
   - import the prepared code into the "User Defined Section "of the testb_lab5.vhd file.

2.7 SIMULATING THE PREPARED VHDL CODE

Simulation process starts by invoking the simulation tool ModelSim by executing,

vsim

which will bring up the ModelSim window, in which specify the work directory as a new library by selecting,

File → New → Library

which will bring up the Create a New Library dialog box, into which enter

Library Name **work**
Library Physical Name **~/fpga/lab/work**
LMB → OK.

which will create the library which will be used for this and all remaining Lab Assignments. Then proceed with the simulation of the testbench module testb_lab5.vhd.

When simulation shows the expected functioning of the architecture shown in Figure 2.1, consider your VHDL source code prepared for the experimentation described in Sections 4.

For the Prelab report, take snapshots of the wave forms around the signal transition time points which indicate the proper functioning of signals.

2.8 SYNTHESIZING THE PREPARED VHDL CODE

Synthesize the prepared top level module **main_lab5.vhd** code to make sure that there are no errors in it, and to generate the configuration file **main_lab5.sof**,

This step completes the Prelab Assignment.

3. NE-2036 FPGA LAB EQUIPMENT

Equipment to be used includes:

- "terasic" proto board DE2, shown in Figure 4.1,
- Mixed-Signal oscilloscope HP54645D,
- SUN Blade-100 computer system,
- USB Blaster cable,
- power supply for the FPGA board DE2.

- HP function generator.

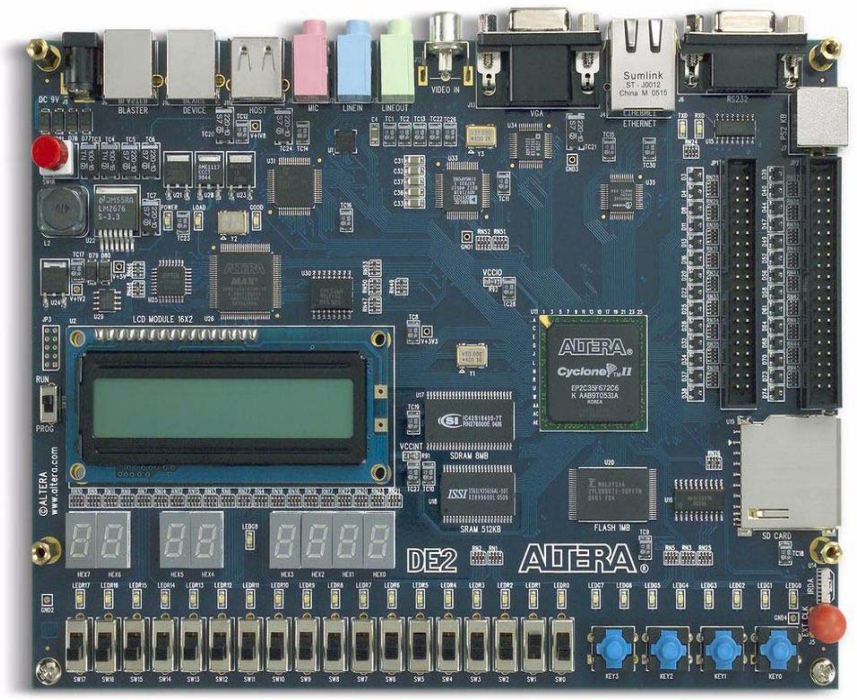

Figure 3.1 View of the terasic's DE2 FPGA protoboard.

4. NE-2036 FPGA LAB ASSIGNMENT

At the beginning of the Lab session, and before anything else should be attempted, a number of connections must be established between the FPGA board and other external equipment. These actions are described in sections 4.1 through 4.2. In the sequel, the sections 4.4 through 4.7 describe the interaction with the FPGA vendor software which leads to the programming of the FPGA and execution of the experimenting with the protoboard circuitry.

4.1 USB BLASTER Connections

The USB Blaster cable must be properly connected to its connector on the DE2 FPGA board port, and to a USB connector on the computer.

4.2 POWER SUPPLY CONNECTIONS

DE2 FPGA proto board is powered by its own external power supply unit which should be connected as follows:

- power supply's 5V DC cord plugged into the DC power jack on the DE2 board,
- power supply's 120V AC cable into the power strip.

4.3 SETTING UP THE APPLICATION SOFTWARE ENVIRONMENT

Since Altera currently does not support USB Blaster drivers for Solaris operating system, this section should be skipped in the Spring 2008 semester!

To log in to a class account in the FPGA Lab, use the computer to which a DE2 FPGA board is connected. Then set up the environment which supports the application software by executing the following commands in the given order:

source /eng/applications/altera/altera.csh
cd lab
mkdir lab5
cd lab5

4.4 ACCESSING THE CONFIGURATION FILE PREPARED IN THE ECC ACCOUNT

Since Altera currently does not support USB Blaster drivers for Solaris operating System, downloading the configuration file light.sof from the ECC student accounts to the DE2 protoboard must be accomplished while DE2 is connected to a Linux or Windows platform. The solution for Spring 2008 Semester is to use a Windows platform located in the NE-1023 Lab. The downloading procedure will be as follows.

4.41 LOGGING ON TO THE WINDOWS PLATFORM

Log in to the Windows platform **L**ocated under the wall-mounted panoramic map of Toledo using:

User name: **fpgauser**
Password: **Altera1**

4.42 MAPPING THE NETWORK DRIVE - SETTING UP THE WINDOWS ENVIRONMENT

Select

RMB → My Computer

which is followed by the appearance of a Map Network Drive dialog box, in which enter:

Map Network Drive **Z:**
Folder: **\\green\<username of the account with configuration file>**

then select,

different user name

after which new dialog box opens up, in which enter

User name: **<student's ECC account user name>**
Password: **<student's ECC account password>**
OK

when the last dialog box closes, select in Map Network Drive dialog box,

Finish

after which the student's home directory will pop up **{in which form?}** on the Windows platform.

4.5 DOWNLOADING THE CONFIGURATION FILE DIRECTLY TO THE FPGA - STEP 7.1 OF THE "TUT_QUARTUS_INTRO_VHDL"

On the computer's desktop, double click on

Quartus II shortcut icon

when the Quartus II window appears, start following Section 7 of the tut_quartus_vhdl tutorial, beginning with the instructions from Section 7.1:

- set the RUN/PROG switch into position,
  RUN
- select
  Tools → Programmer

which opens the Programmer window shown in Figure 41 of the tutorial, in which make all settings described in the last paragraph of page 24 in the tutorial.

At this point, the procedure to be followed deviates from the text in the tutorial because the project, and the process, in which the file light.sof was created took place on the Solaris platform in the NE-1026 lab. Consequently, light.sof will not be listed in the Programmer window, so the next action will be to select,
Add File

which opens the browser window in which navigate to the green mount:
menttest on 'ECC Samba 3.0.25a [green]'Z:

then navigate to the project directory and select,
main_lab5.sof
OPEN

which will open the window Chain1.cdf, in which verify:

- that the proper device code is shown: **EP2C35F672**,
- the box **Program/Configure** is chekmarked,

then select the button,
Start

and observe the stream of messages in the Message window, untill the appearance of the message:
Info: Successfully performed operation

after which proceed with experimenting with the protoboard.

Should the above message fail to appear, the configuration file has some error which must be corrected.

4.6 EXPERIMENTING WITH THE PROTOBOARD CIRCUIT

After a successfull completion of the work described in section 4.5, do the following:

- observe the seven segment display to verify that the counter counts up,
  - when the switch SW[0] is in position "1", and
  - the push button KEY[3] is used to make the counter count,
- observe the LCD pannel to verify that the counter counts down,
  - when the switch SW[0] is in position "0", and
  - the push button KEY[3] is used to make the counter count,,
- as the position of the SW[0] is changed, observe the LEDR[0] and LEDG[7] to verify that they indicate the correct direction of counting.

5. LAB REPORT

To be considered complete the Lab4 report must contain the following,

1. Cover sheet - Lab style, filled out,
2. Description of steps performed in the Pre Lab assignment, and of their outcomes, including a brief description of results of the decoding of instructions which was done in section 2.5.
3. Description of steps performed during the Lab Experiment, and of their outcomes.